ЛАБОРАТОРНАЯ РАБОТА № 2

Архитектура компьютера

Богданюк Анна Васильевна

Содержание

# Цель работы

Целью работы является изучить идеологию и применение средств контроля версий. Приобрести практические навыки по работе с системой git

# Задание

1. Настройка github
2. Базовая настройка git
3. Создание SSH ключа
4. Сознание рабочего пространства и репозитория курса на основе шаблона
5. Сознание репозитория курса на основе шаблона
6. Настройка каталога курса
7. Задание для самостоятельной работы

# Теоретическое введение

Системы контроля версий (Version Control System, VCS) применяются при работе нескольких человек над одним проектом. Обычно основное дерево проекта хранится в локальном или удалённом репозитории, к которому настроен доступ для участников проекта. При внесении изменений в содержание проекта система контроля версий позволяет их фиксировать, совмещать изменения, произведённые разными участниками проекта, производить откат к любой более ранней версии проекта, если это требуется. В классических системах контроля версий используется централизованная модель, предполагающая наличие единого репозитория для хранения файлов. Выполнение большинства функций по управлению версиями осуществляется специальным сервером. Участник проекта (пользователь) перед началом работы посредством определённых команд получает нужную ему версию файлов. После внесения изменений, пользователь размещает новую версию в хранилище. При этом предыдущие версии не удаляются из центрального хранилища и к ним можно вернуться в любой момент. Сервер может сохранять не полную версию изменённых файлов, а производить так называемую дельта-компрессию — сохранять только изменения между последовательными версиями, что позволяет уменьшить объем хранимых данных. Системы контроля версий поддерживают возможность отслеживания и разрешения конфликтов, которые могут возникнуть при работе нескольких человек над одним файлом. Можно объединить (слить) изменения, сделанные разными участниками (автоматически или вручную), вручную выбрать нужную версию, отменить изменения вовсе или заблокировать файлы для изменения. В зависимости от настроек блокировка не позволяет другим пользователям получить рабочую копию или препятствует изменению рабочей копии файла средствами файловой системы ОС, обеспечивая таким образом, привилегированный доступ только одному пользователю, работающему с файлом. Системы контроля версий также могут обеспечивать дополнительные, более гибкие функциональные возможности. Например, они могут поддерживать работу с несколькими версиями одного файла, сохраняя общую историю изменений до точки ветвления версий и собственные истории изменений каждой ветви. Кроме того, обычно доступна информация о том, кто из участников, когда и какие изменения вносил. Обычно такого рода информация хранится в журнале изменений, доступ к которому можно ограничить. В отличие от классических, в распределённых системах контроля версий центральный репозиторий не является обязательным. Среди классических VCS наиболее известны CVS, Subversion, а среди распределённых — Git, Bazaar, Mercurial. Принципы их работы схожи, отличаются они в основном синтаксисом используемых в работе команд.

# Выполнение лабораторной работы

1. Настройка Github Создала учетную запись на Github.(рис.1)

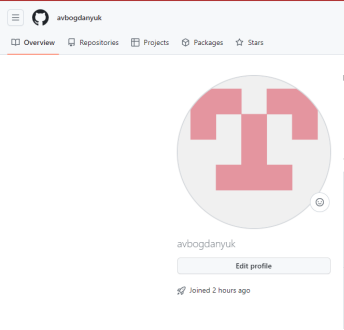


Figure 1: Моя страница на GitHub

1. Базовая настройка git Создаю предварительную конфигурацию git. Открыла терминал и ввела следующие команды, указав имя пользователя и почту владельца репозитории.(рис.2)

Figure 2: Создаю предварительную конфигурацию git

Figure 2: Создаю предварительную конфигурацию git

Настраиваю utf-8 в выводе сообщений git.(рис.3)

Figure 3: Настраиваю utf-8

Figure 3: Настраиваю utf-8

Задаю имя начальной ветки(master).(рис.4)

Figure 4: Задаю имя начальной ветки

Figure 4: Задаю имя начальной ветки

Параметр autocrlf.(рис.5)

Figure 5: Параметр autocrlf

Figure 5: Параметр autocrlf

Параметр safecrlf.(рис.6)

Figure 6: Параметр safecrlf

Figure 6: Параметр safecrlf

1. Создание SSH ключа Генерирую приватный и открытый ключи.(рис.7)

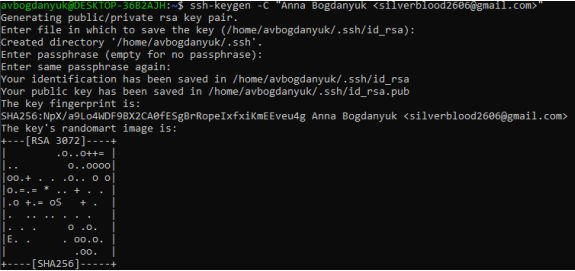


Figure 7: Генерирую приватный и открытый ключи

Вывожу на экран ключ и копирую его.(рис.8)



Figure 8: SSH ключ

Добавляю SSH Key.(рис.9)

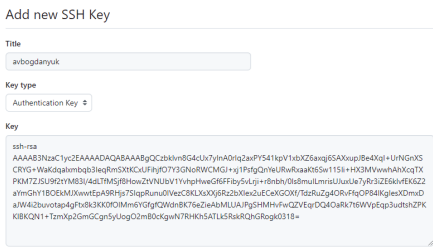


Figure 9: Добавляю SSH ключ

1. Создание рабочего пространства и репозитория курса на основе шаблона Создаю каталог для предмета Архитектура компьютера.(рис.10)

Figure 10: Создаю каталог

Figure 10: Создаю каталог

1. Создание репозитория курса на основе шаблона Создаю репозиторий на основе шаблона.(рис.11)

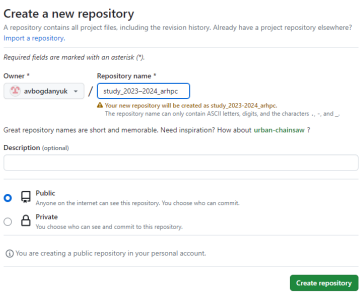


Figure 11: Создаю репозиторий

Клонирую созданный репозиторий.(рис.12)

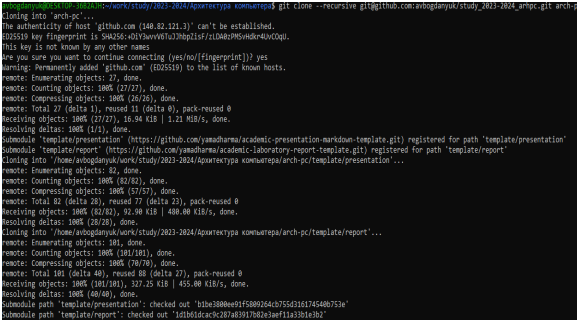


Figure 12: Параметр autocrlf

1. Настройка каталога курса Перехожу в каталог и удаляю там лишние файлы.(рис.13)

Figure 13: Удаляю лишние файлы

Figure 13: Удаляю лишние файлы

Создаю необходимые каталоги.(рис.14)

Figure 14: Создаю каталоги

Figure 14: Создаю каталоги

Отправляю файлы на сервер.(рис.15 и 16)

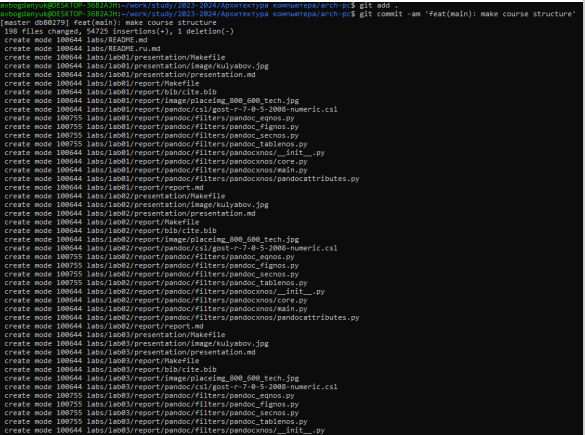


Figure 15: Отправляю файлы на сервер

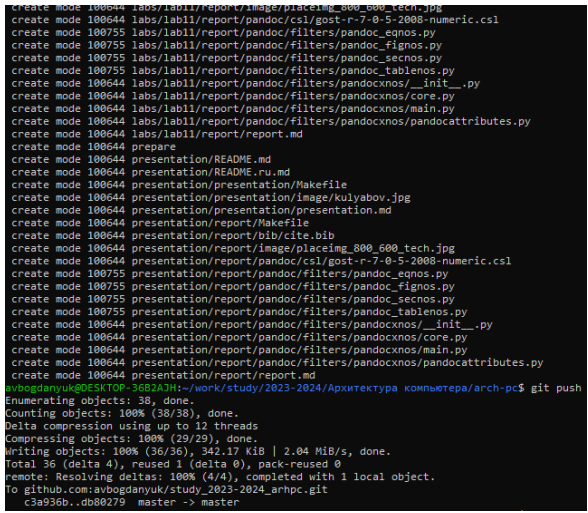


Figure 16: Отправляю файлы на сервер

Проверяю правильность создания иерархии рабочего пространства в локальном репозитории и на странице github.(рис.17)

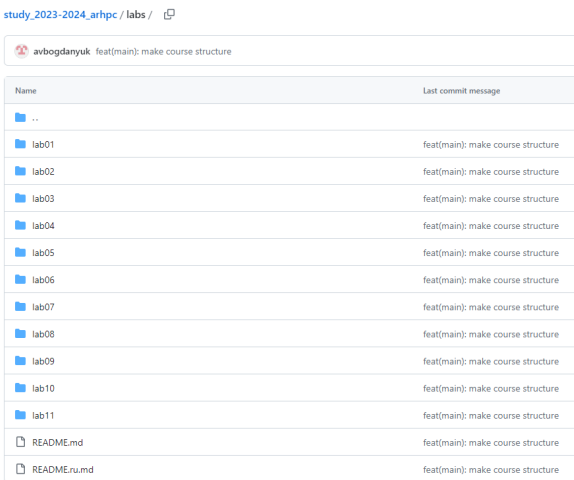


Figure 17: Проверяю правильность создания иерархии рабочего пространства

1. Задание для самостоятельной работы Создаю файл отчета в labs/lab02/report и загружаю его на github. (рис.18 и 19)

Figure 18: Создаю файл отчета

Figure 18: Создаю файл отчета

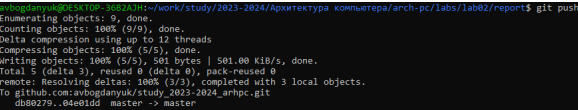


Figure 19: Загружаю его на github

Копирую отчет по лабораторной работе 1 в labs/lab01/report и загружаю его на github.(рис.20 и 21)

Figure 20: Копирую отчет

Figure 20: Копирую отчет

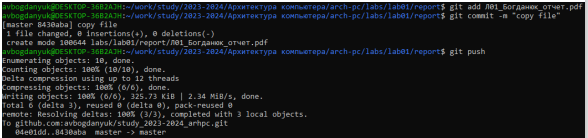


Figure 21: Загружаю его на github

# Выводы

Во время выполнения Лабораторной работы 2 я изучила идеологию и применение средств контроля версий. Приобрела практические навыки по работе с системой git.

# Список литературы